

ĐỀ CƯƠNG CHI TIẾT HỌC PHẦN

1. Tên học phần: Phân tích và Thiết kế hệ thống nâng cao (Advanced System Analysis and Design)

- **Mã số học phần:** CT609

- **Số tín chỉ học phần:** 3 tín chỉ

- **Số tiết học phần:** 30 tiết lý thuyết, 30 tiết thực hành và 60 tiết tự học.

2. Đơn vị phụ trách học phần:

Khoa/Viện/Trung tâm/Bộ môn: Khoa Công nghệ Thông tin & Truyền thông

3. Điều kiện tiên quyết:

- **Điều kiện tiên quyết:** CTH612

- **Điều kiện song hành:**

4. Mục tiêu của học phần:

4.1. Kiến thức:

4.1.1. Hiểu biết thấu đáo về quy trình phân tích thiết kế hướng đối tượng với ngôn ngữ UML.

4.1.2. Phân tích, đánh giá được vai trò của từng loại sơ đồ UML trong quy trình phân tích thiết kế hệ thống hướng đối tượng.

4.2. Kỹ năng:

4.2.1. Khả năng vận dụng thành thực các kỹ thuật thu thập yêu cầu người dùng.

4.2.2. Khả năng mô hình hóa bài toán thực tế sử dụng các sơ đồ UML.

4.2.3. Kỹ năng phát hiện và giải quyết các vấn đề thực tế dựa trên nền tảng máy tính

4.2.4. Kỹ năng phối hợp với các thành viên có chuyên môn khác nhau trong nhóm phân tích thiết kế hệ thống.

4.3. Thái độ/Năng lực tự chủ và trách nhiệm:

4.3.1. Ý thức được ưu, nhược điểm của phương pháp phân tích thiết kế hướng đối tượng.

4.3.2. Ý thức đúng về vai trò của người dùng khi phân tích, thiết kế hệ thống.

5. Mô tả tóm tắt nội dung học phần:

Mục tiêu của học phần là nhằm mang lại cho học viên những kiến thức và kỹ năng về phân tích thiết kế hệ thống hướng đối tượng sử dụng ngôn ngữ UML, kiến thức và kỹ năng có được sẽ chuyên sâu hơn mức trình độ đại học để có thể ứng dụng ngay và hiệu quả vào thực tế đời sống.

Học phần này thuộc khối kiến thức chuyên ngành; sẽ giảng dạy cho học viên các nội dung về cách thức tổ chức đội nhóm, các đặc trưng cơ bản của hướng đối tượng như tính tái sử dụng, tính cơ động và tính tương hợp; các kỹ thuật thực hiện thu thập yêu cầu người dùng; Các kỹ năng, kỹ thuật mô hình hóa ứng dụng thiết kế sơ đồ lớp; Thiết kế phần mềm.

6. Cấu trúc nội dung học phần:

6.1. Lý thuyết

	Nội dung	Số tiết	Mục tiêu
Chương 1. Đội nhóm		5	
1.1.	Tổ chức đội nhóm		
1.2.	Cách tiếp cận dân chủ cho đội nhóm		
1.3.	Cách tiếp cận cổ điển cho lãnh đạo đội nhóm lập trình		4.1.1; 4.2.4
1.4.	Các thành phần khác ngoài lãnh đạo và đội		
1.5.	Đồng bộ và ổn định các đội nhóm		
1.6.	Các đội lập trình chuyên biệt		
Chương 2. Từ các đơn thể (modules) đến các đối tượng (objects)		5	
2.1.	Đơn thể là gì		
2.2.	Sự chặt chẽ		
2.3.	Sự kết nối		
2.4.	Tính bao đóng của dữ liệu		
2.5.	Sự che ẩn thông tin		
2.6.	Sự chặt chẽ và sự móc nối/ kết nối giữa các đối tượng		4.1.1; 4.2.2
2.7.	Cách thức hướng đối tượng (Object-Oriented Paradigm)		
2.8.	Đo lường chất lượng và độ phức tạp của một chương trình		
2.9.	Khó khăn và nguy cơ khi lựa chọn ngôn ngữ lập trình và lập trình hướng đối tượng		
Chương 3. Tính tái sử dụng, tính tương hợp và tính cơ động		5	
3.1.	Kết hợp tính tái khả dụng (reusability) với tính tái sử dụng (reuse) trong công nghệ phần mềm		
3.2.	Khó khăn và nguy cơ khi xem xét công nghệ tái sử dụng được (reusable technology) và hệ thống khách- chủ		
3.3.	Ví dụ minh họa		4.1.1; 4.3.1
3.4.	Đối tượng và sự tái sử dụng		
3.5.	Sự tái sử dụng trong các giai đoạn thiết kế và cài đặt		
3.6.	Mối liên quan giữa tái sử dụng và bảo trì		
3.7.	Tính tương thích/ cơ động/ linh động (Portability)		
3.8.	Các kỹ thuật giúp có được tính tương thích		
3.9.	Các khuynh hướng (trends) về tính tương hợp		

Chương 4. Thu thập yêu cầu	5
4.1. Phân tích lĩnh vực	
4.2. Điểm bắt đầu của các dự án phân mềm	
4.3. Định nghĩa bài toán và phạm vi	
4.4. Định nghĩa và phân loại yêu cầu	4.1.1; 4.2.3;
4.5. Các kiểu sưu liệu về yêu cầu (requirements document)	4.3.2
4.6. Rà soát lại các yêu cầu	
4.7. Quản trị sự thay đổi các yêu cầu	
4.8. Khó khăn và nguy cơ khi phân tích lĩnh vực và yêu cầu	
Chương 5. Mô hình hóa	5
5.1. Sơ đồ lớp	
5.2. Mẫu thiết kế (Design patterns)	4.1.1; 4.1.2;
5.3. Thiết kế đặt con người làm trọng tâm (User-centered design)	4.2.2
5.4. Mô hình hóa các tương tác và hành vi	
Chương 6. Tạo kiến trúc và thiết kế phần mềm	5
6.1. Quy trình thiết kế	
6.2. Các nguyên tắc để dẫn đến thiết kế tốt	
6.3. Các kỹ thuật để lấy quyết định đúng cho thiết kế	
6.4. Phát triển hướng về mô hình (MDD: Model Driven Development)	4.2.2; 4.2.3
6.5. Kiến trúc phần mềm (Software architecture)	
6.6. Các mẫu về kiến trúc (Architectural patterns)	
6.7. Tạo một sưu liệu tốt cho thiết kế	
6.8. Khó khăn và nguy cơ khi thiết kế	

6.2. Thực hành

Nội dung	Số tiết	Mục tiêu
Bài 1. Thiết kế sơ đồ hoạt vụ	5	
1.1. Các khái niệm		4.2.1; 4.2.2;
1.2. Biểu diễn yêu cầu người dùng với sơ đồ hoạt vụ		4.2.3
1.3. Mô tả trường hợp sử dụng		
Bài 2. Thiết kế sơ đồ lớp	10	
2.1. Các khái niệm		
2.2. Thiết kế lớp thực thể		4.2.1; 4.2.2;
2.3. Thiết kế lớp biên		4.2.3
2.4. Thiết kế lớp điều khiển		
2.5. Từ sơ đồ lớp đến mã chương trình		
Bài 3. Thiết kế sơ đồ tuần tự	5	
3.1. Các khái niệm		4.2.1; 4.2.2;
3.2. Quy trình thiết kế sơ đồ tuần tự		4.2.3
3.3. Kiểm thử sơ đồ tuần tự		
Bài 4. Thiết kế sơ đồ hoạt động, sơ đồ trạng thái	10	
4.1. Các khái niệm		4.2.1; 4.2.2;
4.2. Sơ đồ hoạt động		4.2.3
4.3. Sơ đồ trạng thái		

4.4 Sơ đồ hoạt động vs Sơ đồ trạng thái

7. Phương pháp giảng dạy:

- Giảng viên hướng dẫn lý thuyết trên lớp
- Học viên tổ chức thành nhiều nhóm (mỗi nhóm tối đa 4 học viên) thực hiện thảo luận nhóm, thiết kế các sơ đồ theo yêu cầu với đặc tả hệ thống do giảng viên cung cấp.
- Các nhóm góp ý, phản biện các bản thiết kế của nhóm bạn mình.

8. Nhiệm vụ của học viên:

Học viên phải thực hiện các nhiệm vụ như sau:

- Tham dự tối thiểu 80% số tiết học lý thuyết.
- Tham gia đầy đủ 100% giờ thực hành/thí nghiệm/thực tập và có báo cáo kết quả.
- Thực hiện đầy đủ các bài tập nhóm/bài tập và được đánh giá kết quả thực hiện.
- Tham dự kiểm tra giữa học kỳ.
- Tham dự thi kết thúc học phần.
- Chủ động tổ chức thực hiện giờ tự học.

9. Đánh giá kết quả học tập của học viên:

9.1. Cách đánh giá

Học viên được đánh giá tích lũy học phần như sau:

TT	Điểm thành phần	Quy định	Trọng số	Mục tiêu
1	Điểm bài tập nhóm	- Báo cáo/thuyết minh - Được nhóm xác nhận có tham gia	20%	4.2.1; 4.2.4
2	Điểm kiểm tra giữa kỳ	- Thi viết (60 phút)	20%	4.1.1; 4.1.2; 4.2.2; 4.2.3
3	Điểm thi kết thúc học phần	- Thi viết (90 phút) - Tham dự đủ 80% tiết lý thuyết và 100% giờ thực hành - Bắt buộc dự thi	60%	4.1.1; 4.1.2; 4.2.2; 4.2.3

9.2. Cách tính điểm

- Điểm đánh giá thành phần và điểm thi kết thúc học phần được chấm theo thang điểm 10 (từ 0 đến 10), làm tròn đến một chữ số thập phân.
- Điểm học phần là tổng điểm của tất cả các điểm đánh giá thành phần của học phần nhân với trọng số tương ứng. Điểm học phần theo thang điểm 10 làm tròn đến một chữ số thập phân, sau đó được quy đổi sang điểm chữ và điểm số theo thang điểm 4 theo quy định về công tác học vụ của Trường.

10. Tài liệu học tập:

Thông tin về tài liệu

Số đăng ký cá biệt

[1] Phạm Thị Xuân Lộc, Trương Quốc Định, Phan Tấn Tài, Giáo trình phân tích hệ thống hướng đối tượng. Nxb. Đại học Cần Thơ, 2014.

CNTT.002987 –
CNTT.002991

MOL.073679-
MOL.073683

- [2] Grady Booch, Robert Maksimchuk, Michael Engle, Jim Conallen, Kelli Houston, Ph.D. Young, Bobbi, Object-Oriented Analysis and Design with Applications. ISBN-13: 978-0201895513
ISBN-10: 020189551X
- [3] Timothy C. Lethbridge, Robert Laganière, Object-Oriented software Engineering: Practical software development using UML and Java, McGraw-Hill Education, 2005. ISBN-13: 978-0077109080
ISBN-10: 0077109082
- [4] Bhuvan Unhelkar, Practical object oriented design, Australia: Thomson, 2005. CNTT.001326

Số thứ tự trên kệ sách (số phân loại): 005.117 / U57

11. Hướng dẫn học viên tự học:

Tuần	Nội dung	Lý thuyết (tiết)	Thực hành (tiết)	Nhiệm vụ của học viên
1	Chương 1: Đội nhóm 1.1. Tổ chức đội nhóm 1.2. Cách tiếp cận dân chủ cho đội nhóm 1.3. Cách tiếp cận cổ điển cho lãnh đạo đội nhóm lập trình 1.4. Các thành phần khác ngoài lãnh đạo và đội 1.5. Đồng bộ và ổn định các đội nhóm 1.6. Các đội lập trình chuyên biệt	5	0	- Nghiên cứu trước: + Tài liệu [3]: chương 1, 2 + Tài liệu [4]: chương 1, 2
2	Chương 2: Từ các đơn thể (modules) đến các đối tượng (objects) 2.1. Đơn thể là gì 2.2. Sự chặt chẽ 2.3. Sự kết nối 2.4. Tính bao đóng của dữ liệu 2.5. Sự che ẩn thông tin 2.6. Sự chặt chẽ và sự móc nối/ kết nối giữa các đối tượng 2.7. Cách thức hướng đối tượng (Object-Oriented Paradigm) 2.8. Đo lường chất lượng và độ phức tạp của một chương trình 2.9. Khó khăn và nguy cơ	5	0	- Nghiên cứu trước: + Tài liệu [2]: chương 3 + Tài liệu [3]: chương 1, 2 + Tài liệu [4]: chương 1, 2

	<p>khí lựa chọn ngôn ngữ lập trình và lập trình hướng đối tượng</p>			
3	<p>Chương 3: Tính tái sử dụng, tính tương hợp và tính cơ động</p> <p>3.1. Kết hợp tính tái khả dụng (reusability) với tính tái sử dụng (reuse) trong công nghệ phần mềm</p> <p>3.2. Khó khăn và nguy cơ khi xem xét công nghệ tái sử dụng được (reusable technology) và hệ thống khách-chủ</p> <p>3.3. Ví dụ minh họa</p> <p>3.4. Đối tượng và sự tái sử dụng</p> <p>3.5. Sự tái sử dụng trong các giai đoạn thiết kế và cài đặt</p> <p>3.6. Mối liên quan giữa tái sử dụng và bảo trì</p> <p>3.7. Tính tương thích/ cơ động/ linh động (Portability)</p> <p>3.8. Các kỹ thuật giúp có được tính tương thích</p> <p>3.9. Các khuynh hướng (trends) về tính tương hợp</p>	5		<p>- Nghiên cứu trước: + Tài liệu [3]: chương 3</p>
4	<p>Chương 4: Thu thập yêu cầu</p> <p>4.1. Phân tích lĩnh vực</p> <p>4.2. Điểm bắt đầu của các dự án phần mềm</p> <p>4.3. Định nghĩa bài toán và phạm vi</p> <p>4.4. Định nghĩa và phân loại yêu cầu</p> <p>4.5. Các kiểu sưu liệu về yêu cầu (requirements document)</p> <p>4.6. Rà soát lại các yêu cầu</p> <p>4.7. Quản trị sự thay đổi các yêu cầu</p> <p>4.8. Khó khăn và nguy cơ khi phân tích lĩnh vực và yêu cầu</p>	5		<p>- Nghiên cứu trước: + Tài liệu [3]: chương 4</p>
5-9	<p>Chương 5. Mô hình hóa</p> <p>5.1. Sơ đồ lớp</p> <p>5.2. Mẫu thiết kế (Design patterns)</p> <p>5.3. Thiết kế đặt con người</p>	5	30	<p>- Nghiên cứu trước: + Tài liệu [1] + Tài liệu [4]: chương 3, 4 và 5 + Tài liệu [3]: chương 5, 8 + Tài liệu [2]: chương 5</p>

	làm trọng tâm (User-centered design) 5.4. Mô hình hóa các tương tác và hành vi			
10	Chương 6: Tạo kiến trúc và thiết kế phần mềm 6.1. Quy trình thiết kế 6.2. Các nguyên tắc để dẫn đến thiết kế tốt 6.3. Các kỹ thuật để lấy quyết định đúng cho thiết kế 6.4. Phát triển hướng về mô hình (MDD: Model Driven Development) 6.5. Kiến trúc phần mềm (Software architecture) 6.6. Các mẫu về kiến trúc (Architectural patterns) 6.7. Tạo một sơ liệu tốt cho thiết kế 6.8. Khó khăn và nguy cơ khi thiết kế	5		- Nghiên cứu trước: + Tài liệu [3]: chương 9

Cần Thơ, ngày ... tháng ... năm 2019

**TL. HIỆU TRƯỞNG
TRƯỞNG KHOA**

GIẢNG VIÊN BIÊN SOẠN

Nguyễn Hữu Hòa

TS. Trương Quốc Định